

# Expectation Maximization

# Expectation-Maximization

## E-M

If the underlying governing pdf is known only in its general form, and there may or may not be missing data as well, we need E-M

- To reconstruct the underlying pdf
- To find missing data based on the underlying pdf

# Example: Missing Normally Distributed Data

- We have 4 data points from a Gaussian distribution with unit variance. 2 data points are missing.
  - 5
  - 11
  - $x$
  - $x$

- We need to infer the parameters of the density from which the observations were drawn.
- We are given one of the parameters in this case, variance=1
- We seek the most likely  $\mu$  for the Gaussian density function; *i.e.* we seek  $\mu_{ML}$
- Once we can find the mean of the underlying Gaussian distribution ( $\mu_{ML}$ ), we can generate the missing data

We can take advantage of properties of Gaussian densities to make life easier


- The value (arg) of  $\mu$  that yields the least square error

$$\sum_{i=1}^m (x_i - \mu)^2$$

is simply the sample mean

$$\frac{1}{n} \sum_{i=1}^n x_i$$

So, to maximize the likelihood estimate of the parameter  $\mu_{ML_j}$  it is necessary and sufficient to use the sample mean for  $\mu$

$$\mu_{ML} = \arg \min_{\mu} \sum_{i=1}^m (x_i - \mu)^2$$


# Strategy

- Guess the parameter
- Expectation: Find the expected data, given the parameter
- Maximization: Find the most likely parameter, given the data, by finding the sample mean
- Repeat and converge to a solution for  $\mu_{ML}$

# Example: Missing Normally Distributed Data

- The first step is to guess at a mean for the pdf. Guess 0. The expectation of a normally distributed value *is* the mean. (expectation step)
  - 5
  - 11
  - x
  - x



Now minimize the error of the data estimate by using the expected mean (maximization step)

5  
11  
0  
0



Now re-estimate the mean by using the data (expectation step)

$$(5=11+0+0)/4=4$$

Now minimize the error of the data estimate by using the new expected mean (maximization step)

5

11

4

4

Now re-estimate a new mean by using the data (expectation step)

$$(5+11+4+4)/4=6$$

Now minimize the error of the data estimate by using the new expected mean (maximization step)

5

11

6


6

Now re-estimate a new mean by using the data (expectation step)

$$(5+11+6+6)/4=7.5$$

Now minimize the error of the data estimate by using the new expected mean (maximization step)

5  
11  
7.5  
7.5



Now re-estimate a new mean by using the data (expectation step)

$$(5+11+7.5+7.5)/4=7.75$$

# $\mu_{\text{ML}}$ values for subsequent iterations

- 7.75
- 7.875
- 7.9375
- 7.96875
- 7.984375
- 7.992188
- 7.996094
- 7.998047
- 7.999023
- 7.999512
- 7.999756

**The iteration is guaranteed to converge to a local minimum!**

# Types of Problems for E-M

- Parametric
  - We need to find the parameters of the pdf
- Non Parametric (Most Common)
  - We need to find pdf itself

# Parametric Problem Example

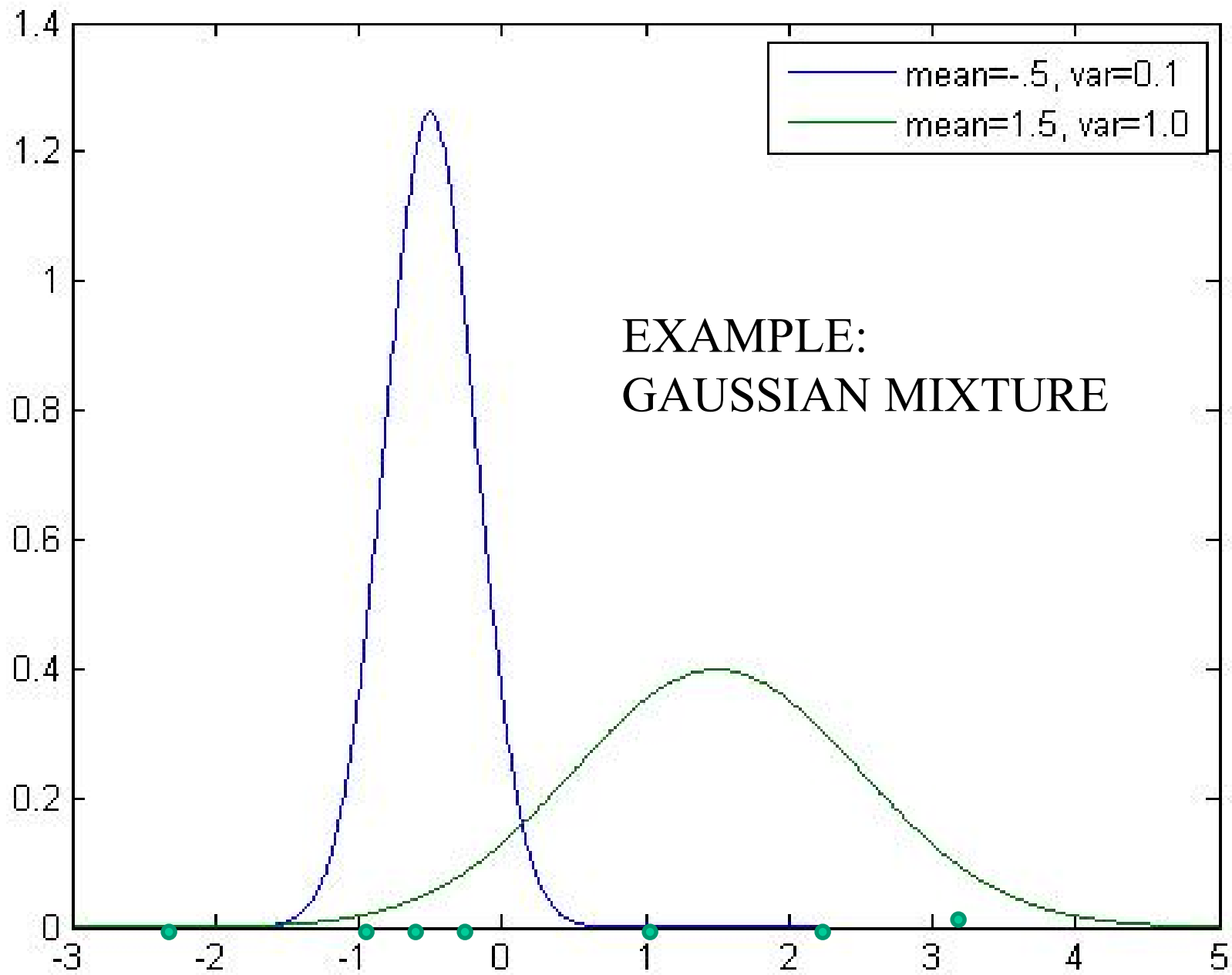
(After Mitchell, Machine Learning)

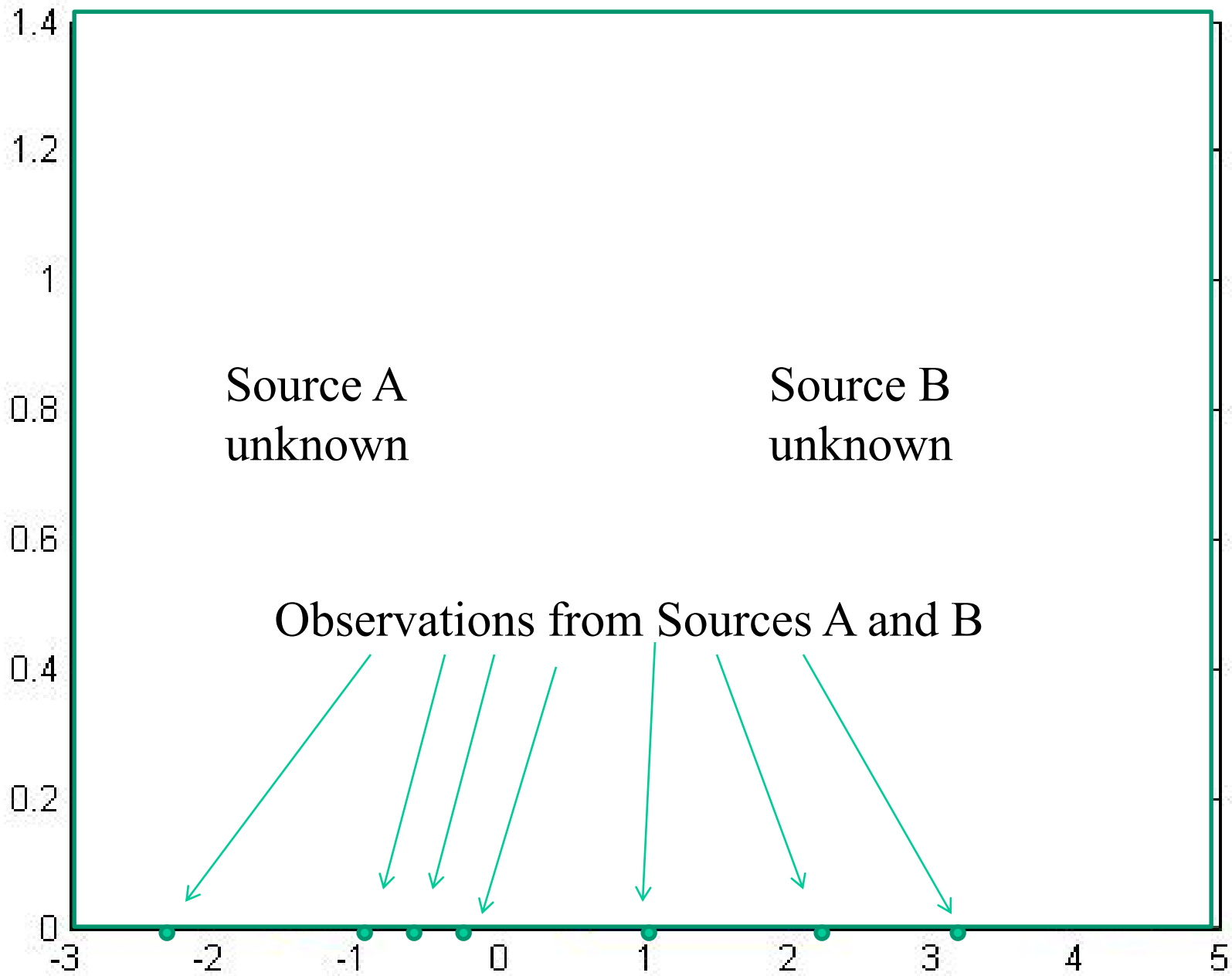
# Parametric Example

There are 2 Gaussian processes, mixing.  
We don't see the processes nor do we know the parameters. (In the example, for simplicity we will assume unit variance for each, so only the means are the unknown parameters)

We have observations, but we don't know about the two specific processes to which the observations belong







We need to infer the  
parameters of the densities from  
which the observations were  
drawn

# EM

Typically, we have data which are generated by multiple, say WLOG, 2, Gaussian probability densities, A and B.

The data are  $a_1, a_2, a_3, \dots, b_1, b_2, \dots$ , but the difficulty is that we only observe the unlabeled data  $x_1, x_2, x_3, \dots$ , where the random variable  $x$  could be either an  $a$  or a  $b$

We need to figure out the parameters  $\Theta$  of the entire system which parameters of A and B that are most likely to have generated the data. If the system is Gaussian, we are looking for  $\mu$  and  $\sigma^2$  for each of the generating densities.

# EM: The Problem Statement

As stated, we are give the observations= $x_1, x_2, x_3$  where the  $x$  are unlabeled data from A,B and we are looking for the parameter(s) of  $\Theta$

Then the labels that associate  $x$  with either  $a$  or  $b$  are the hidden data.

# A Mystery

We don't know the labels

But..

If we knew the *parameters* of each of the mother densities, we could calculate the probability that each  $x$  came from A and the probability that it came from B, and choose the more probable.

# A Solution

We have only the data. How can we know what's what?

Answer: Assume there are labels for each data point  $x_i$ . These labels,  $z_{i,1}, z_{i,2}, \dots, z_{i,j}$ , are associated with all  $j$  mother densities (in this case,  $j=2$ ), each telling the probability that the  $j^{\text{th}}$  density generated the data point

# Expectation Maximization

So... We need a 2 step process

1. Determine the expectations of the hidden variables, given the parameters  
 $p(\text{Data}|\text{Parameters})$
2. Determine the best parameters, given the data

$p(\text{Parameters}|\text{Data})$

This is the *likelihood* of the data and we wish to *maximize this likelihood*



# EM: In Summary

- We need to optimize the expression

$$P(O_{bs}, H_{idden} | \Theta)$$

Given only

$$P(O_{bs} | \Theta)$$

by choosing the best  $\Theta$

# Getting Started

While we don't know the initial values of the parameters, we can *guess*

If the guess is reasonable, this process can be shown to converge to a local maximum likelihood (Dempster,Laird,Rubin)

# EXAMPLE

## The Gaussian Mixtures already presented

Let the data set consist of a set of points  
 $[(x_i, z_{1i}, z_{2i})]$

Here, the observed data point is  $x$  and the hidden data are the  $z$ 's. The  $z_{i,j}$  represent the probability that  $x_i$  came from the  $j^{\text{th}}$  density

Given 2 processes A, B (W.L.O.G.)

Probability that  $x$  was generated from  
a process with mean A

Expectation  $z_A =$

---

Probability that  $x$  was generated from  
either process with mean A or process  
with mean B

Q: What is the probability of any  $x_i$  given process  $(\mu, \sigma^2)_j$ ?

Answer for this particular model (Gaussian Mixture):

$$p(x_i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^2}$$

# EXPECTATION Step

$$E[z_{i,j}] = \frac{p(x = x_i \mid \mu = \mu_j)}{\sum_{n=1}^2 p(x = x_i \mid \mu = \mu_n)}$$



Notice we are normalizing

Since it is a Gaussian Distribution, we can define the probabilities concisely

$$E[z_{i,j}] = \frac{e^{-\frac{1}{2\pi\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^2 e^{-\frac{1}{2\pi\sigma^2}(x_i - \mu_n)^2}}$$

The denominator effectively 'normalizes' the numerator, making it a probability measure

# MAXIMIZATION STEP

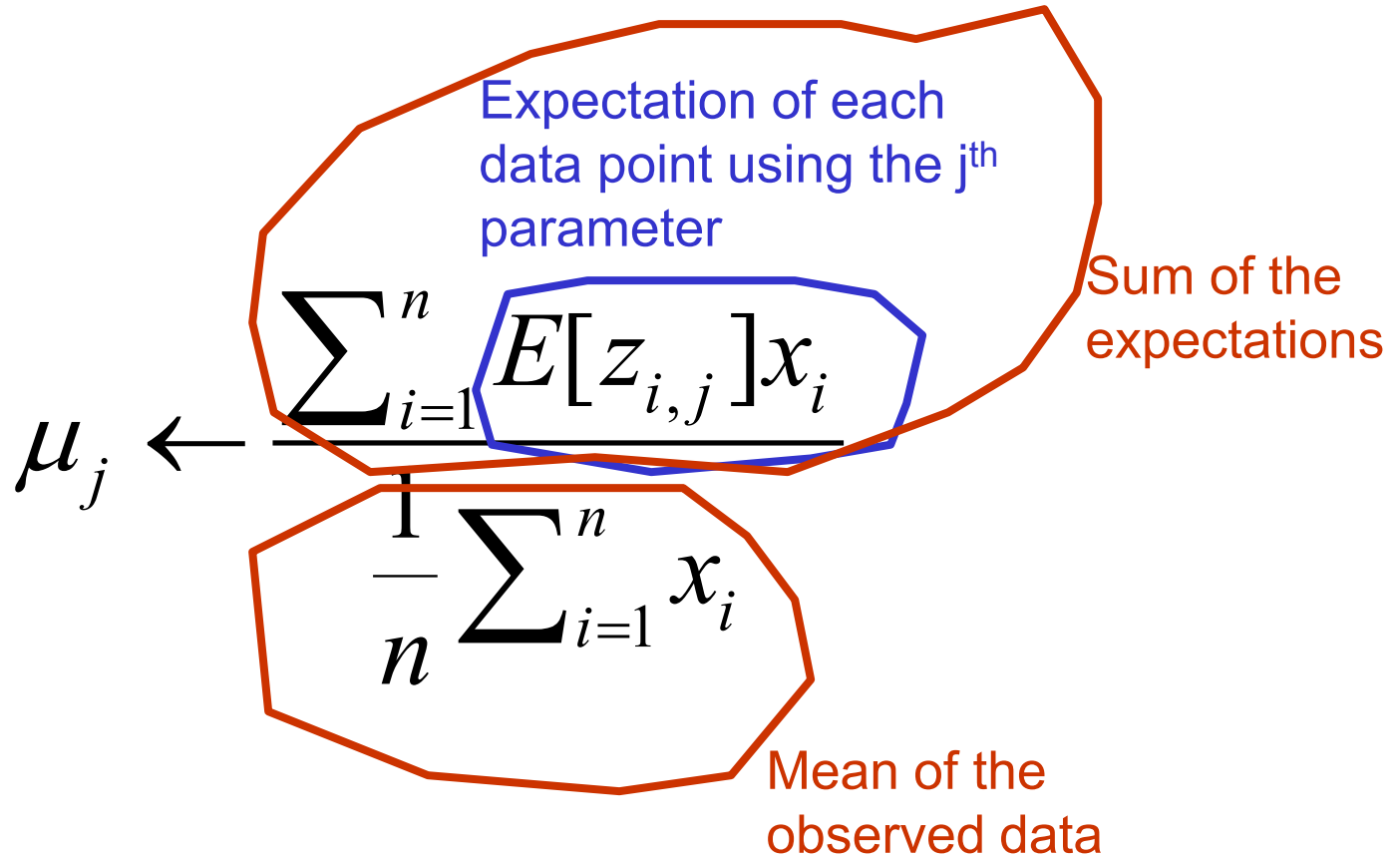
So, the Maximum Likelihood estimate of the  $j^{\text{th}}$  parameter ( $j^{\text{th}}$  mean) is the normalized sum of the value of each sample weighted by the probability of each sample

$$\mu_j \leftarrow \frac{\sum_{i=1}^n E[z_{i,j}] x_i}{\frac{1}{n} \sum_{i=1}^n x_i}$$



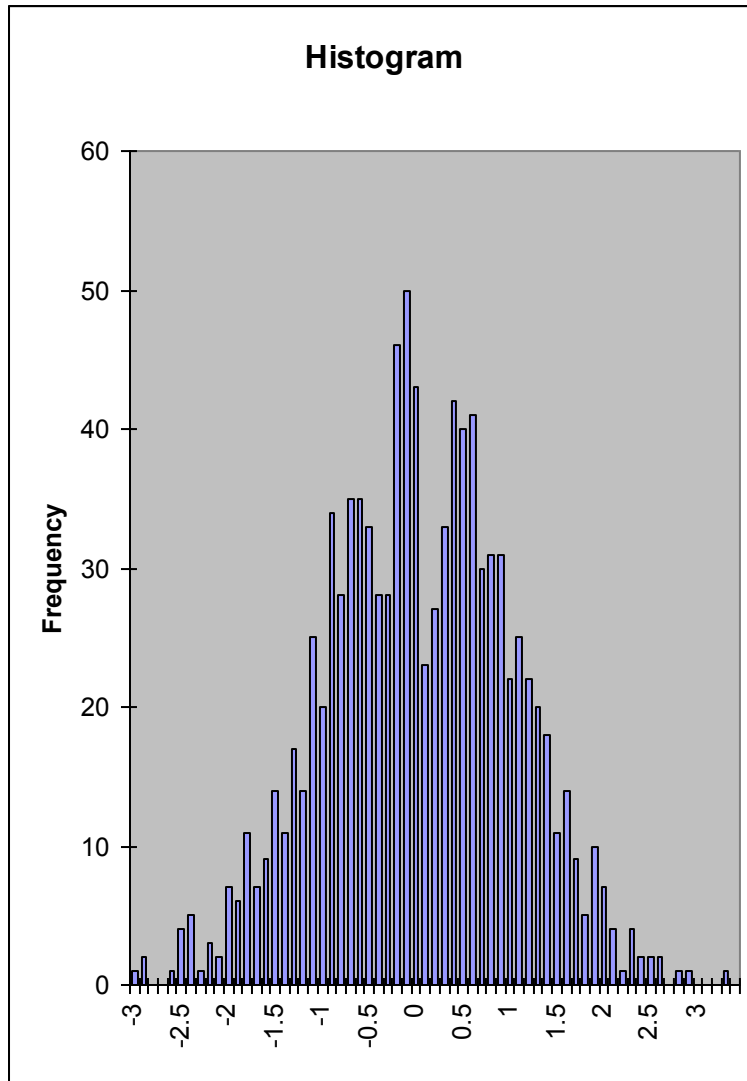
# MAXIMIZATION STEP-explained

For the the  $j^{\text{th}}$  parameter ( $j^{\text{th}}$  mean) :



# DEMO

[http://bioinformatics.uchc.edu/Bioinformatics\\_tools/EMdemo.asp](http://bioinformatics.uchc.edu/Bioinformatics_tools/EMdemo.asp)



This is a single sample of 1000 points drawn from the random number generator used in the Demo

# Non Parametric Example

After Karp, R University of Washington

# The Task

We observe blood types in a bunch of people

- These types are A,B, AB, and O
- (the Blood types determined by the Blood Bank are the *phenotypes*)
- The phenotypes are the *observed* data

## The Task:

Infer the frequencies ( *ie* a discrete pdf) of the blood type alleles A,B and O, using known principles of genetics, by means of the hidden data

# What's missing?

We need to know the genotypes which underlie the phenotypic expression

- The possible genotypes are AA, AO, OA, AB, BB, BO, OB, OO
- These genotypes are the *hidden* data

# Experiment: Observe Phenotypes

- Determine the blood type of 30 people

Sample:

- Type A 16
- Type B 2
- Type AB 1
- Type O 11

# Create a matrix for the allele probabilities

Start with a guess at the probabilities

'left' allele	'right' allele	Initial guess
$p(A)$	$p(A)$	.4
$p(B)$	$p(B)$	.2
$p(O)$	$p(O)$	.4

$$\begin{pmatrix} p(A) & p(A) \\ p(B) & p(B) \\ p(O) & p(O) \end{pmatrix} = \begin{pmatrix} .4 & .4 \\ .2 & .2 \\ .4 & .4 \end{pmatrix}$$

The goal is to refine the guess into a most likely estimate

Now create a matrix for the hidden data (the genotypes) based on the first guess of the allele probabilities

$$H = \begin{matrix} & \begin{matrix} AA & AO & OA & BB & BO & OB & AB & BA & OO \end{matrix} \\ \begin{pmatrix} .16 & .16 & .16 & & & & & & \\ & & & .04 & .08 & .08 & & & \\ & & & & & & .08 & .08 & \\ & & & & & & & & .16 \end{pmatrix} & \begin{matrix} \text{prob of genotypes determining phenotype A} \\ \text{prob of genotypes determining phenotype B} \\ \text{prob of genotypes determining phenotype AB} \\ \text{prob of genotypes determining phenotype O} \end{matrix} \end{matrix}$$



# EXPECTATION

NORMALIZE each row so that the row entries represent a probability.

Specifically, each entry in a row represents the probability that the entry's genotype (column heading) led to the entry's phenotype (row heading).

# Matrix normalized

$$H = \begin{matrix} & \begin{matrix} AA & AO & OA & BB & BO & OB & AB & BA & OO \end{matrix} \\ \begin{matrix} .33 & .33 & .33 \\ & .2 & .4 & .4 \\ & & & .5 & .5 \\ & & & & & & & & 1 \end{matrix} \end{matrix}$$

*Note: Red annotations in the original image show the calculation of the first row's values:*

- $.16$  is divided by  $.16 + .16 + .16$  to yield  $.33$ .
- $.08$  is divided by  $.04 + .08 + .08$  to yield  $.4$ .

# MAXIMIZATION

- Now we wish to recover the probabilities of the alleles being drawn from the population.  
(Remember this was the goal)
- The Probability matrix was set up as a 2 column matrix
  - Column 1 was the probability of recovering the ‘left’ allele from the population\*
  - Column 2 was the probability of recovering the ‘right’ allele from the population

\*In this particular example, the left and right allele probabilities are equal

# MAXIMIZATION

Now, ask the question: Which entries have contributed to a specific left allele, say, A?

(We will ask and answer the same question for all alleles, both left and right)

Answer: For the 'left' A, it is

- row 1 (A phenotype) col 1 (genotype AA)
- row 1 (A phenotype) col 2 (genotype AO),
- row 3 (AB phenotype), col 7 (AB genotype)

$$H = \begin{pmatrix}
AA & AO & OA & BB & BO & OB & AB & BA & OO \\
.33 & .33 & .33 & & & & & & \\
& & & .2 & .4 & .4 & & & \\
& & & & & & .5 & .5 & \\
& & & & & & & & 1.
\end{pmatrix}
\begin{matrix}
A \\
B \\
AB \\
O
\end{matrix}$$

# MAXIMIZATION

- So, compute the probability of the Left A allele as follows:
  - Sum of entries in row 1 (A phenotype)  
 $.333 + .333 = .666$   
There are 16 individuals with phenotype A
  - Sum of entries in row 3 (AB phenotype) = .5  
There is one individual with phenotype AB

# MAXIMIZATION

- $P(A_{\text{left}}) = .667 \times 16 + 0.5 \times 1 = 11.06$
- Do the same calculation for  $A_{\text{right}}$ , for  $B_{\text{right}}$ , for  $O_{\text{right}}$ , and for  $O_{\text{left}}$

Continuing the calculation for all alleles....

	LEFT	RIGHT
A	11.06	11.06
B	1.7	1.7
O	16.75	16.75

then normalizing.....

	LEFT	RIGHT
A	0.375	0.375
B	0.058	0.058
O	0.568	0.568



# MAXIMIZATION

We thus recover the ‘new’ allele probability matrix

This was our goal

$$\begin{pmatrix} p(A) & p(A) \\ p(B) & p(B) \\ p(O) & p(O) \end{pmatrix} = \begin{pmatrix} .375 & .375 \\ .058 & .058 \\ .568 & .568 \end{pmatrix}$$

# EM

Now, use the new Allele Probability  
Matrix and  
ITERATE!

Quit when it converges.

Demonstration:

[http://bioinformatics.uhc.edu/Bioinformatics\\_tools/EMDemo\\_alleles.html](http://bioinformatics.uhc.edu/Bioinformatics_tools/EMDemo_alleles.html)

Many applications of EM, but we are interested in three.

- Learning transition probabilities in a Hidden Markov Model, given only the emissions (observations) as training data. There is an efficient (polynomial) special-case algorithm, the Baum-Welch Algorithm, exploiting the Markov structure of the HMM .
- Learning a most likely motif subsequence given many sequences of data, each of which contains a coding subsequence for the protein function of interest. The algorithm is called the Multiple-Sequence Expectation-Maximization Motif Elicitation (MEME)
- Unsupervised cluster discovery